

SYSTEMS, METHODS, AND COMPUTER PROGRAM PRODUCTS TO INTEGRATE USER-DEFINED OPERATIONS INTO A DATABASE TRANSACTION

CROSS-REFERENCE TO RELATED PATENT

5

In US Patent Number 5,953,719, entitled "Heterogeneous Database System with Dynamic Commit Procedure Control," assigned to the assignee of the present invention, and incorporated herein in its entirety by this reference, there is described a heterogeneous database system employing a dynamic commit procedure control.

10 Although not limited thereto, the present invention employs such a method in one of its preferred embodiments.

BACKGROUND OF THE INVENTION

15 Field of the Invention

The present invention is directed to the field of computer-implemented database transaction management. It is more particularly directed to efficiently extending database transaction semantics to include user-defined operations.

20

Description of the Background Art

A computer-implemented database is a collection of data, organized in the form of tables. A table typically consists of columns that represent data of the same nature, and records that represent specific instances of data associated with the table. A
25 relational database is a database that may be a set of tables containing information that is manipulated in accordance with the relational model associated with the data. For example, the product marketed under the trademarks IBM DB2 stores the data associated with the database in tables, and each table has a name. It will be appreciated that other
30 vendors also provide databases, including relational databases.

Many databases enable users to extend database operations with user-customizable features. Examples of such extensibility mechanisms include user-defined

functions and operations, stored procedures, extenders, blades, wrappers, and connectors. Such extensibility mechanisms will be referred to herein as user-defined operations.

Such user-defined operations may be implemented by program code used to perform tasks that are specific to applications that operate in conjunction with databases such as data computations and transformations, and to retrieve data from external sources. The user-defined operations are often tightly coupled with the database, and typically may be executed as part of normal database operations such as queries and updates, and performed on data managed by the database before it is returned to the application. This tightly coupled execution model is crucial to ensuring high performance execution of data-intensive database applications.

A transaction is a unit of work that often changes data and that is typically created by operations associated with a database application program. A transaction typically includes execution of an application program specified sequence of operations that are initiated with a begin transaction operation, that include one or more update or read access operations, and that end with either a commit or a rollback operation. Database transactions associated with highly critical database applications operate reliably by ensuring that transactions are completely performed when committed or fully rolled back if an error occurs during the operation of the transaction. User-defined operations may perform updates, so it is desirable to be able to extend the operation of database transactions to include the user-defined operations.

A highly reliable transaction typically includes the following features that ensure reliable updates of the associated data: atomicity, consistency, isolation, and durability (ACID). Atomicity ensures that the results of a transaction are treated as a unit and either completed or backed out as a unit, either by a data commit or roll back operation. Those skilled in the art will appreciate the data commit operation and the data roll back operation. Atomicity is typically implemented by a single-phase commit protocol in a non-distributed database environment and by a two-phase commit protocol in a distributed database environment. The two-phase commit protocol typically probes the computer-related resources associated with the transaction to ensure they are ready to

commit and only commits the transaction if all resources are ready to commit.

Consistency ensures that the transaction changes data from one valid state to another valid state. Isolation ensures that intermediate changes to the data that occur during the transaction operation are not accessible until the transaction has completed a data commit operation. Durability ensures that intermediate changes to the data that occur during the transaction survive subsequent computer-system failures.

While most database products do not extend transaction semantics to user-defined operations, the IBM DB2 product does enable such extended transaction semantics through integration with the Resource Recovery Services Transaction Control Facility embedded in the operating system marketed under the trademark Z/OS®. This integration enables user-defined operations to commit and rollback through the database, but not as part of the database transaction, and relies on a proprietary solution.

It would be useful for database products to extend non-proprietary highly reliable database transaction semantics, such as computer transaction protocols, to include user-defined operations that access resources external to the database ("external resources"). More particularly it would be useful if user-defined operations operated in existing, well-defined, highly reliable, industry standard database transaction protocol semantics that support atomicity. From the foregoing it will be apparent that there is still a need to improve the techniques associated with highly reliable transaction protocol semantics that support user-defined operations and that are in databases.

SUMMARY OF THE INVENTION

An embodiment of the present invention relates to systems, methods, and computer products that efficiently support techniques associated with highly reliable transaction protocol semantics in databases. More particularly, the preferred embodiment of the present invention extends database transaction semantics that support a single-phase or a two-phase commit protocol to include user-defined operations that access external resources by means of a novel use of the XA X/Open® Distributed Transaction

Processing Protocol ("XA Transaction Protocol"). It will be appreciated by those skilled in the art that XA is an industry standard two-phase commit application programming interface (API) that operates between a transaction manager, such as an XA transaction manager and a resource manager. The phrase, "distributed transaction processing" herein
5 refers to transaction processing distributed over computer operating systems, such as the product marketed under the trademark UNIX and the IBM OS/390®. Those skilled in the art will appreciate that an XA transaction manager manages distributed transactions by means such as, coordinating decisions about commit or rollback of pending transactions, and coordinating failure recovery. It has been difficult in the past for database products
10 to extend highly reliable database transaction semantics to user-defined operations that support industry-standards and atomicity.

The preferred embodiment of the present invention novelly enables user-defined operations that access external resources that support the XA Transaction Protocol to be
15 included in database transactions by enabling the database to act as an XA Transaction Manager. More particularly and in the preferred embodiment of the present invention, user-defined operations are novelly included in a data access module (DAM) as described in US Patent 5,953,719. A DAM is a program which specifies procedures and provides necessary data to enable data transactions to be carried out with a respective database
20 type. An entry in a DAM may comprise a default commit protocol that is either single phase or two phase, and the commit protocol may be used with a respective database type. While the preferred embodiment of the present invention uses DAM technology to register the XA Transaction Protocol Interface, it will be appreciated that other embodiments within the spirit of the invention may be implemented without such a
25 DAM.

Further, the preferred embodiment of the present invention novelly ensures atomicity for the results of the database transaction that includes both the user-defined operations and database operations. Such operations may access external resources
30 managed by distributed transaction processing resource managers or by local database

managers. It will be appreciated by those skilled in the art that resource managers typically access computer system resources, such as databases and message systems.

5 The preferred embodiment of the present invention also supports an industry-standard management of database transactions, the XA Transaction Protocol. It will be appreciated that many applications and products are XA-compliant implementations, such as message systems and application services. Therefore, since the preferred embodiment of the present invention supports the XA Transaction Protocol, such XA-compliant implementations may be easily integrated to take advantage of the user-defined
10 operations.

According to the preferred embodiment of the present invention, the database acts in part as an XA transaction manager, and novelly translates and executes distributed transactions that include database operations and user-defined operations that access
15 resources that adhere to the XA Transaction Protocol. Further, the user-defined operations enable user access to a variety of XA resources, such as a message system, by the XA Resource Manager. More particularly, the preferred embodiment of the present invention enables the user to invoke, via the user-defined operations, software code in the resource manager that may, in combination with the DAM, novelly support the XA
20 Transaction Protocol. Such user-defined operations software code is directed to a computer resource, such as a message system. Therefore, the preferred embodiment of the present invention novelly enables user-defined operations to access critical computer resources that are integrated with database resources while ensuring data recoverability via atomicity in database transactions.

25

An embodiment of the present invention is achieved by systems, methods, and computer products that extend highly reliable database transaction semantics, by including user-defined operations in the database transaction. An embodiment of the method comprises two phases, a Registration Phase and a Runtime Phase. During the
30 Registration Phase, the user enables access to the user-defined operation in the application program code and registers it with the database that acts as an XA transaction

manager. Further, the user registers, with the database, the XA Transaction Protocol Interfaces that are used by the user-defined operation code

During the Runtime Phase, the user initiates a database transaction and invokes the user-defined operation. The application program or an external transaction manager may initiate the database transaction. The invocation of the user-defined operation is recorded with the database thereby triggering, as part of the normal transaction processing during commit or rollback, the database to invoke the registered interfaces to the XA Transaction Protocol that are, in the preferred embodiment of the present invention, implemented in the DAM. Further, and by communication between the XA Resource Manager and the DAM that is described in US Patent 5,953,719, the database acts as an XA transaction manager to coordinate two phase commit with the database and the external resources accessed by the user-defined operation.

An embodiment of the present invention novelly improves highly reliable database transactions by extending database transaction semantics to user-defined operations using the non-proprietary XA Transaction Protocol. Other aspects and advantages of the present invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram that illustrates the computer systems that may implement the present invention;

Figure 2 is a block diagram that illustrates the preferred embodiment of the present invention;

Figure 3 is a block diagram that illustrates a database transaction;

Figure 4 is a flow diagram that illustrates the method of the preferred embodiment of the present invention; and

Figure 5 is a block diagram of a computer system suitably configured for employment of the present invention.

5

In the following detailed description and in the several figures of the drawings, like elements are identified with like reference numerals.

DETAILED DESCRIPTION OF THE INVENTION

10

As shown in the drawings and for purposes of illustration, an embodiment of the invention relates to systems, methods, and computer products that efficiently support techniques associated with highly reliable transaction protocol semantics in databases. More particularly, the preferred embodiment of the present invention extends database transaction semantics that support a single-phase or a two-phase commit protocol to include user-defined operations that access external resources by means of a novel use of the XA X/Open® Distributed Transaction Processing Protocol ("XA Transaction Protocol").

15

20

Figure 1 is a block diagram that illustrates the computer systems that may operate with the present invention. As shown in Figure 1 and in element 100, the preferred embodiment of the present invention may operate in a networked computer system configuration. Therefore, a client computer system 102 may communicate with a server computer system 104 during the operation of the present invention. The preferred embodiment of the present invention operates in the server computer system 104.

25

The application program 120 operates in the client computer system 102 or the server computer system 104 to perform the present invention. For example, information may be communicated to either the server 104 or the client 102 via the user interface 117, such as an invocation to execute the application program 120. The user may include user-defined query code 124 in the application program 120 that is subsequently used to

30

invoke user-defined operations software code 210. Such code supports the XA Transaction Protocol 214 and is directed to a computer system resource 226, such as a database 110. By supporting the XA Transaction Protocol 214, a user-defined operation 210 that accesses the computer system resource 226 that adheres to the XA Transaction protocol 214 may be novelly included in a database transaction 228. Elements 210, 214, 226 and 228 are described in detail with reference to Figure 2

The user interface 117 may communicate with the preferred embodiment of the present invention, either via batch input 119 or user input 118. Further, the database 110 may be configured in the memory 558 of the client 102 or the server 104. Alternatively the database 110 may be configured in computer storage such as that of a disk 122. Element 558 is described with reference to Figure 5.

Figure 2 is a block diagram that illustrates the preferred embodiment of the present invention. Therefore the user interface 117 enables a user to access an application program 120 that in turn either begins a database transaction 228 with the database 110 or accesses the XA transaction manager 208 to do so. The database acting as an XA transaction manager 229 typically includes an XA Transaction Protocol 214 that is novelly used in the preferred embodiment of the present invention to enable user-defined operations 210 to be included in database transactions 228. Therefore, the user-defined query code 124 invokes user-defined operation code 210 that is included with other transaction-related code, such as database operations 220, to generate commands used in the database transaction 228. Those skilled in the art will appreciate that invoking user-defined operation code 210 may be implemented by a variety of techniques, such as providing a pointer or procedural call to a software library that contains the actual user-defined operation code 210.

Further, the preferred embodiment of the present invention novelly registers user-defined operations code 210 with the database 110 that communicates with an XA Resource Manager 224 or a local resource manager 221, so that the user-defined operations 210 may be invoked by the user-defined query code 124. In the preferred

embodiment of the present invention, such communications are facilitated by including a Registered XA Transaction Protocol Interface 206 in the DAM 227.

By including the user-defined operation code 210 along with computer system software operations, such as database operations 220, during rollback or commit of the database transaction 228 the preferred embodiment of the present invention novelly ensures atomicity. Such operations may access distributed transaction processing resource managers, such as XA resource managers 224, in addition to local resource managers 221. Such resource managers typically access computer system resources 226, such as databases 110, message systems 225, and file systems 223. Any such computer system resources 226 may be accessed by the user-defined operations 210, and may be located on a local computer resource 100 or a distributed computer resource 100. Element 100 is described with reference to Figure 1.

The preferred embodiment of the present invention novelly operates with the DAM 227 to enable the database transaction 228 to operate with both database operations 220 and user-defined operations 210. More particularly and in the present example, the database acting as an XA Transaction Manager for user defined operations 229 translates and executes distributed database transactions 228 that include database operations 220 and user-defined operations 210 that may access distributed computer system resources 226 or local computer system resources 226. Therefore, the preferred embodiment of the present invention enables the database 110 to operate as an XA transaction manager 229 when managing user-defined operations 210. More particularly, the database acting as an XA Transaction Manager 229 invokes the XA Transaction Protocol 214 to operate with the XA Resource Manager 224 that manages the computer system resources 226 accessed by user-defined operations 210. Therefore, user-defined operation code 210 and database operations code 220 may be included in a database transaction 228 that ensures atomicity.

Those skilled in the art will appreciate that structured query language (SQL) is a standardized language for defining and manipulating data in a relational database 110. While SQL command lines are used herein by means of example it should not be viewed

as limiting. Many other computer-accessible forms of program code may be used to implement embodiments of the present invention.

Refer now to Figure 3 that illustrates a database transaction 228 according to the preferred embodiment of the present invention. A begin transaction 322 initiates the database transaction 228. The application program 120 or the transaction manager 208 that is external to the database 110 may initiate the database transaction 228. In the present example an SQL SELECT command locates name and address information from an employee table 324. Then an SQL UPDATE command updates the employee salary information 326. Now, a user defined operation 210 in the present example initiates a command to send a message to a message queue 328. Then the related table is updated via an SQL UPDATE command 330. Finally the transaction is ended 332. Elements 110 and 120 are described with reference to Figure 1, and elements 208, 210, and 228 are described with reference to Figure 2.

Figure 4 is a flow diagram that illustrates an embodiment of the present invention that includes two phases, a registration phase and a runtime phase. Therefore, as shown in element 401, the registration phase is started. As shown in element 402, the user provides the user-defined operation code 210 by the application program code 120 and registers it with the database that acts as an XA Transaction Manager 229. The user-defined operations 210 may use the XA Transaction Protocol Interface 206, which is included in the DAM 227. Further as shown in element 404, the user registers, with the database 110, the XA Transaction Protocol Interfaces 206, which are used by the user-defined operations 210. Then the registration phase is ended 408. Elements 110 and 120 are described with reference to Figure 1, and elements 206, 210, 227, and 229 are described with reference to Figure 2.

The runtime phase is initiated, as shown in element 410. During the Runtime Phase, and as shown in element 412, a database transaction 228 is initiated. In the preferred embodiment of the present invention and as shown in element 413, the user-defined operation 210 is incorporated in the database transaction 228 by support from the

DAM 227. As shown in element 414, the invocation of the user-defined operation 210, via the user-defined query code 124, is recorded with the database 110 thereby triggering, as part of the normal transaction processing during commit or rollback, the database XA Resource Manager 224 to invoke the user registered XA Transaction Protocol Interfaces 206 by use of the DAM 227. Using the registered interfaces to the XA Transaction Protocol 214, the database supports inclusion and coordination of the user-defined operations 210 in the database transaction 228. Further, and by communication between the database 229 and the DAM 227, the database acts as an XA transaction manager 229 to coordinate two phase commit of the computer system resources 226 accessed by the user-defined operations 210. The database transaction 228 that includes the user-defined operations 210 is completed, as shown in element 416. The runtime phase is then ended, as shown in element 418. Element 124 is described with reference to Figure 1, and elements 214, 226 and 228 are described with reference to Figure 2.

Figure 5 is a block diagram of a computer system 500, suitable for employment of the present invention. System 500 may be implemented on a general-purpose microcomputer, such as one of the members of the IBM Personal Computer family, or other conventional workstation or graphics computer devices. In its preferred embodiment, system 500 includes a user interface 505, a user input device 510, a display 515, a printer 520, a processor 555, a read only memory (ROM) 550, a data storage device 122, such as a hard drive, a random access memory (RAM) 540, and a storage media interface 535, all of which are coupled to a bus 525 or other communication means for communicating information. Although system 500 is represented herein as a standalone system, it is not limited to such, but instead can be part of a networked system, including a wireless system. The computer system 500 may be connected locally or remotely to fixed or removable data storage devices 122 and data transmission devices 545. For example, the server computer system 104 and the client computer system 102 also could be connected to other computer systems 500 via the data transmission devices 545. Elements 102 and 104 are described with reference to Figure 1.

The RAM 540, the data storage device 122 and the ROM 550, are memory

components 558 that store data and instructions for controlling the operation of the processor 555, which may be configured as a single processor or as a plurality of processors. The processor 555 executes a program 542, such as the application program 120 to perform the methods of the present invention, as described herein. Element 120 is
5 described with reference to Figure 1.

While the program 542 is indicated as loaded into the RAM 540, it may be configured on a storage media 530 for subsequent loading into the data storage device 122, the ROM 550, or the RAM 540 via an appropriate storage media interface 535.
10 Storage media 530 can be any conventional storage media such as a magnetic tape, an optical storage media, a compact disk, or a floppy disk. Alternatively, storage media 530 can be a random access memory 540, or other type of electronic storage, located on a remote storage system.

15 Generally, the computer programs and operating systems are all tangibly embodied in a computer usable device or medium, such as the memory 558, the data storage device 122, or the data transmission devices 545, thereby making an article of manufacture, such as a computer program product, according to the invention. As such, the terms "computer program product" as used herein are intended to encompass a
20 computer program accessible from any computer usable device or medium.

Moreover, the computer programs 542 and operating systems are comprised of instructions which, when read and executed by the server computer system 104 and the client computer system 102, cause the server computer system 104 and the client
25 computer system 102 to perform the steps necessary to implement and use the present invention. Under control of the operating system, the computer programs 542 may be loaded from the memory 558, the data storage device 122, or the data transmission devices 545 into the memories 558 of the server computer system 104 and the client computer system 102 for use during actual operations.

30

User interface 505 is an input device, such as a keyboard or speech recognition

subsystem, for enabling a user to communicate information and command selections to the processor 555. The user can observe information generated by the system 500 via the display 515 or the printer 520. The user input device 510 is a device such as a mouse, track-ball, or joy stick that allows the user to manipulate a cursor on the display 515 for communicating additional information and command selections to the processor 555.

Those skilled in the art will recognize many modifications may be made to this configuration without departing from the scope of the present invention. When operating in accordance with one embodiment of the present invention, the system efficiently extends database transaction semantics that support a single-phase or a two-phase commit protocol to include user-defined operations 210 while supporting the XA Transaction Processing Protocol 214. The processor 555 and the program 542 collectively operate as a module, such as the application program 120, for extending database transaction semantics. It will be appreciated that the present invention offers many advantages over prior art techniques. Elements 210 and 214 are described with reference to Figure 2.

The present invention is typically implemented using one or more computer programs, each of which executes under the control of an operating system and causes the server computer system 104 and the client computer system 102 to perform the desired operations as described herein. Thus, using the present specification, the invention may be implemented as a machine, process, method, system, or article of manufacture by using standard programming and engineering techniques to produce software, firmware, hardware or any combination thereof.

It will be understood that various alternatives and modifications may be devised by those skilled in the art. However, these should not be viewed as limitations upon the practice of these teachings, as those skilled in the art, when guided by the foregoing teachings, may derive other suitable characteristics of a similar or different nature. The present invention is intended to embrace all such alternatives, modifications and variances that fall within the scope of the appended claims

Trademarks

- IBM, DB2, and OS/390 are trademarks of International Business Machines Corporation in the United States, other countries, or both. X/Open is a trademark or a registered trademark of the Open Group. UNIX is a trademark or a registered trademark of UNIX System Laboratories, Inc.
- 5